37181 DISCRETE MATHEMATICS

Lecture 8: correctness of computer code; PMI and WOP

PLAN

- correctness of computer code
- PMI and WOP

CORRECTNESS OF COMPUTER CODE

We say a procedure/computer program/(algorithm) is correct if

- It stops after a finite number of steps..
- The output claimed to be produced by the algorithm is what is promised.

CORRECTNESS OF COMPUTER CODE

We say a procedure/computer program/(algorithm) is correct if

- It stops after a finite number of steps..
- The output claimed to be produced by the algorithm is what is promised.

Wikipedia: In computer science, a *loop invariant* is a property of a program loop that is true before (and after) each iteration.

It is a logical assertion, sometimes checked within the code by an assertion call. Knowing its invariant(s) is essential in understanding the effect of a loop.

CORRECTNESS OF COMPUTER CODE

Here is a fragment of slightly useless code.

```
int j = 9;
for(int i=0; i<10; i++)
   j--;
```

There is no output, but we will use this to illustrate loop invariant. Something that is true at the start, and remains true after each iteration, so is true at the end also.

Here is a fragment of slightly useless code.

```
int j = 9;
for(int i=0; i<10; i++)
   j--;
```

There is no output, but we will use this to illustrate loop invariant. Something that is true at the start, and remains true after each iteration, so is true at the end also.

Termination:

Loop invariant:

Here is a fragment of slightly useless code.

```
int j = 9;
for(int i=0; i<10; i++)
   j--;
```

There is no output, but we will use this to illustrate loop invariant. Something that is true at the start, and remains true after each iteration, so is true at the end also.

Termination:

Loop invariant: $i + j =$

CORRECTNESS OF COMPUTER CODE

```
input x,d positive integers;
q=0;
r=x;
while(r>=d)
  r=r-d;
  q++;
return (q,r)
```

```
input x,d positive integers;
q=0;
r=x;
while(r>=d)
  r=r-d;
  q++;
return (q,r)
```

Termination:

Loop invariant:

EXAMPLE FROM WIKIPEDIA

```
int max(int n,const int a[]) {
    int m = a[0];
    // m equals the maximum value in a[0...0]
    int i = 1;
    while (i != n) {
        // m equals the maximum value in a[0...i-1]
        if (m < a[i])
            m = a[i];
        // m equals the maximum value in a[0...i]
        ++i;
        // m equals the maximum value in a[0...i-1]
    }
    // m equals the maximum value in a[0...i-1], and i==n
    return m;
}
```

Termination:

Loop invariant:

CORRECTNESS OF COMPUTER CODE

Euclidean algorithm: $a, b \in \mathbb{Z}_+$ (for simplicity) and $a \neq 0 \vee b \neq 0$.

The steps are:

1. Start with (a, b) such that $a \geqslant b$. (ie. put them in order).
2. While $b \neq 0$,

   compute the remainder $0 \leqslant r < b$ of a divided by b.

   set $a = b, b = r$ (and thus $a \geqslant b$ again).
3. Return a

Euclidean algorithm: $a, b \in \mathbb{Z}_+$ (for simplicity) and $a \neq 0 \vee b \neq 0$.

The steps are:

1. Start with (a, b) such that $a \geqslant b$. (ie. put them in order).
2. While $b \neq 0$,
   compute the remainder $0 \leqslant r < b$ of a divided by b.
   set $a = b, b = r$ (and thus $a \geqslant b$ again).
3. Return a

Termination:

Loop invariant:

More practice on loop invariants in the homework and worksheet.

Finally, so far in this course, we have asked you to *accept* two “facts” or axioms:

WOP:

PMI:

Axiom: true without following from any other fact.

Theorem

WOP implies PMI

Proof.

Assume $P(0)$ and $(P(k) \to P(k+1))$ are both true. Define

$$S = \{i \in \mathbb{N} \mid P(i) \text{ is false}\}.$$

$\square$

Theorem

PMI implies WOP

Proof.

$\square$

NEXT

Next lecture:

- Relations
- Functions
- one-to-one
- onto
- bijection